

DEEP LEARNING COM TENSORFLOW



REDE NEURAL CONVOLUTIVA

DIEGO RODRIGUES DSC

INENET

CRONOGRAMA

Dia	Aula	Trab
02/09	Workshop de Deep Learning	
04/09	Deep FeedForward	
09/09	Rede Neural Convolutiva	Modelo Baseline
11/09	AutoEncoder	
16/09	Representation & Transfer Learning	Modelo Profundo
18/09	Sequências	
23/09	Modelos Generativos	Deployment
25/09	Apresentação dos Trabalhos Parte II	



REDE NEURAL CONVOLUTIVA

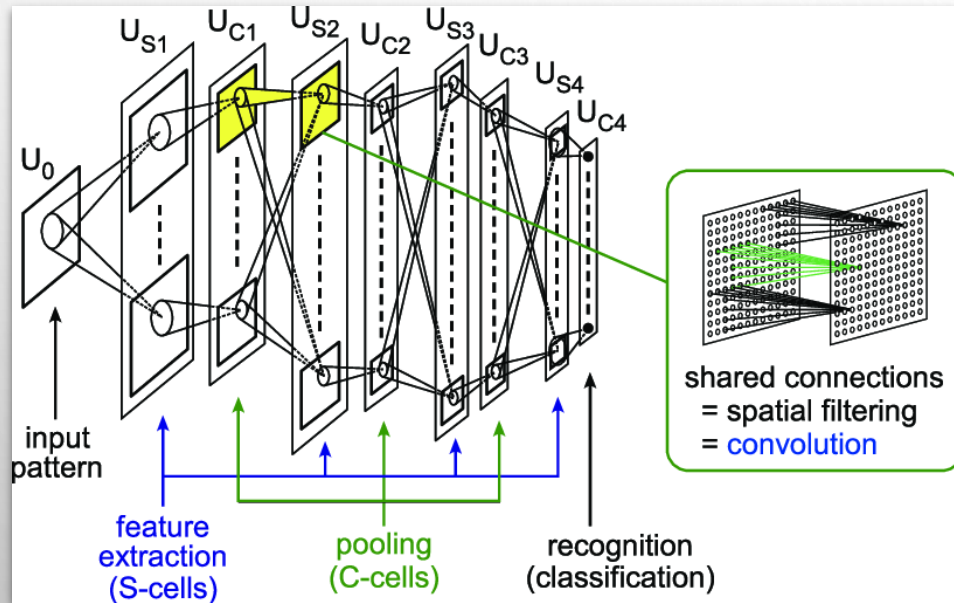
- PARTE 1 : TEORIA
 - VISÃO COMPUTACIONAL
 - BUSINESS UNDERSTANDING
 - MODELING
 - REDE NEURAL CONVOLUTIVA
 - MAX POOLING
- PARTE 2 : PRÁTICA
 - CNN DÍGITOS
- PARTE 3 : TRABALHOS
 - SEMANA 1

PARTE 1 : TEORIA

The background is a light gray gradient. In the top-left and bottom-right corners, there are several realistic-looking water droplets of various sizes, some overlapping. In the top-center, there is a faint, circular, embossed-like pattern that resembles a stylized sun or a flower.

VISÃO COMPUTACIONAL

1980'S - NEUROCOGNITRON



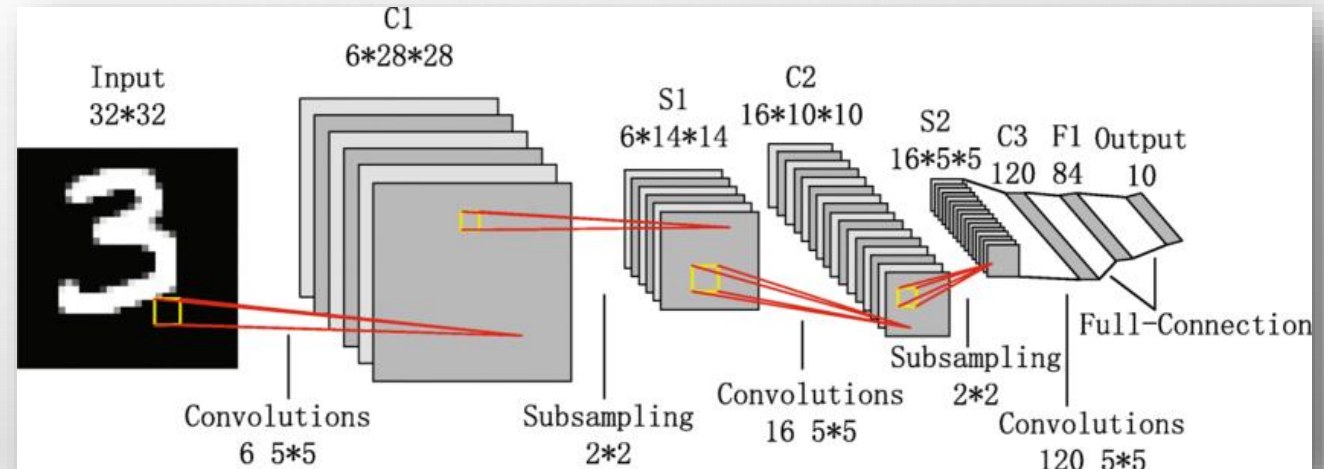
Em paralelo a Marr, o cientista da computação Kunihiro Fukushima desenvolveu uma rede neural que era **capaz de reconhecer padrões em imagens**. A rede, chamada de **Neurocognitron**, foi a primeira **Rede Neural Convolucional (CNN)** da história.

1998 – LENET-5

Criada por **Yann LeCun** para reconhecimento de dígitos escritos à mão (usada no dataset MNIST).

LeNet foi uma das primeiras arquiteturas de CNN e introduziu conceitos como camadas convolutivas e de pooling.

Estrutura simples: poucas camadas convolutivas seguidas de camadas totalmente conectadas, usada para problemas de classificação de imagens pequenas (28x28 pixels)



2010 - IMAGENET



Em 2010 o dataset “ImageNet” foi disponibilizado, junto com as competições anuais de classificação de imagem. Essa base proveu a fundação para os modelos de CNN utilizados atualmente.

2012 - ALEXNET

Desenvolvida por Alex Krizhevsky, foi um marco ao vencer a competição ImageNet com grande margem.

Introduziu redes mais profundas e complexas (8 camadas) e o uso de ReLU como função de ativação, acelerando o treinamento.

Usou técnicas como Dropout e Data Augmentation para reduzir overfitting.

Demonstrou o poder de GPUs para treinamento de redes profundas, revolucionando o campo de deep learning.

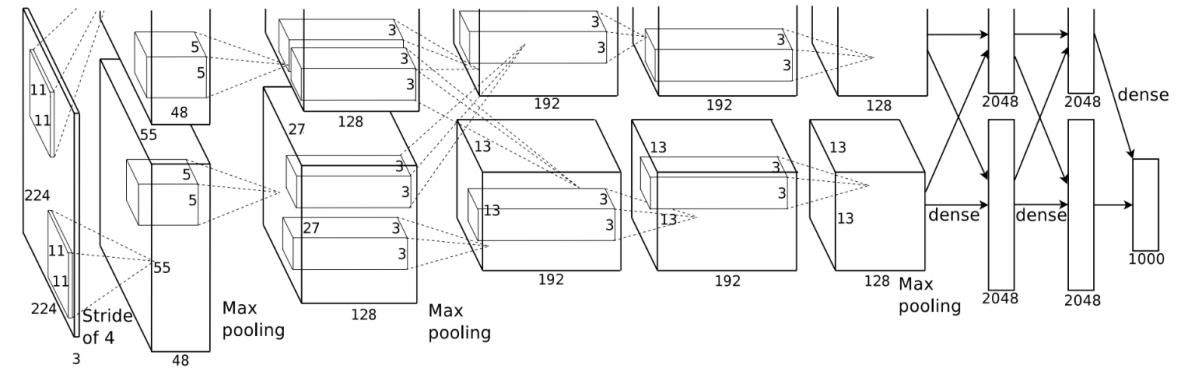


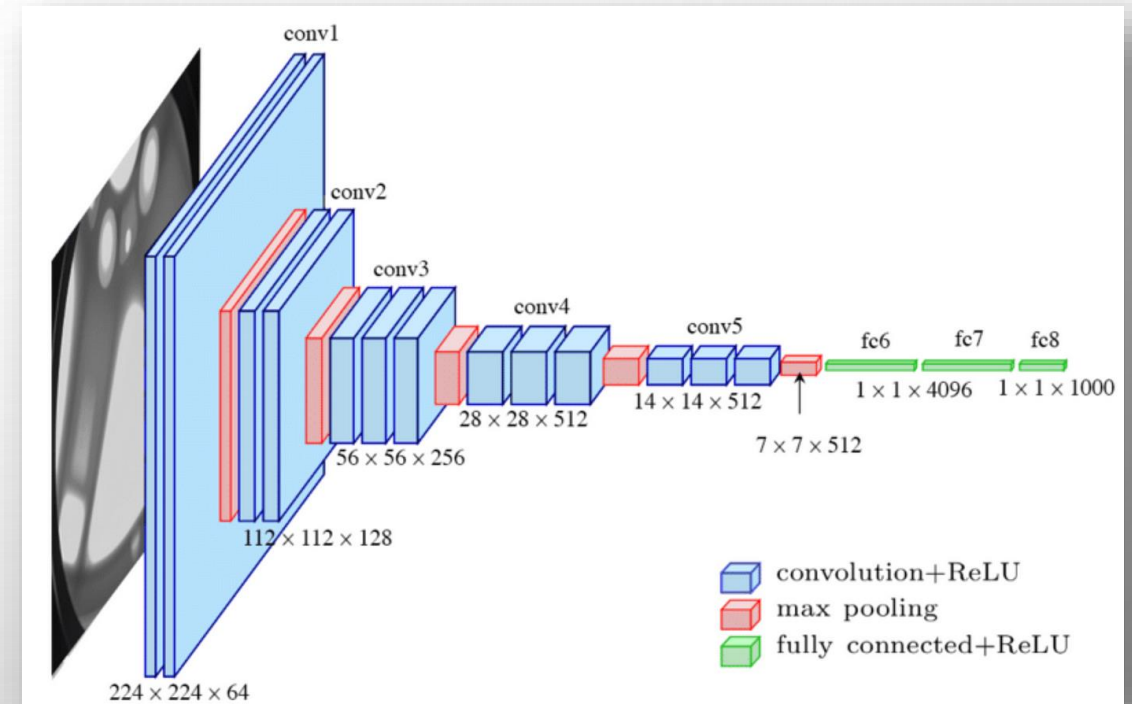
Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

2014 - VGG

Criada por Karen Simonyan e Andrew Zisserman, a arquitetura VGG popularizou o uso de múltiplas camadas convolutivas empilhadas com filtros pequenos (3×3).

O modelo VGG-16 e VGG-19 (16 e 19 camadas, respectivamente) mostraram que redes mais profundas podem capturar representações mais complexas das imagens.

Simple em estrutura, mas com muitas camadas, é amplamente usada como base em transfer learning.

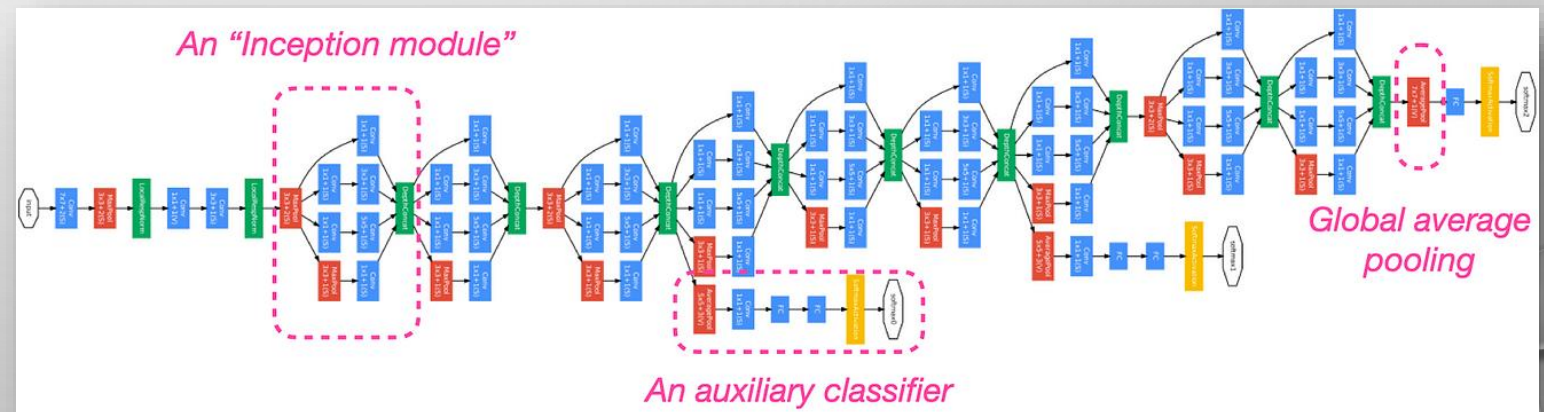
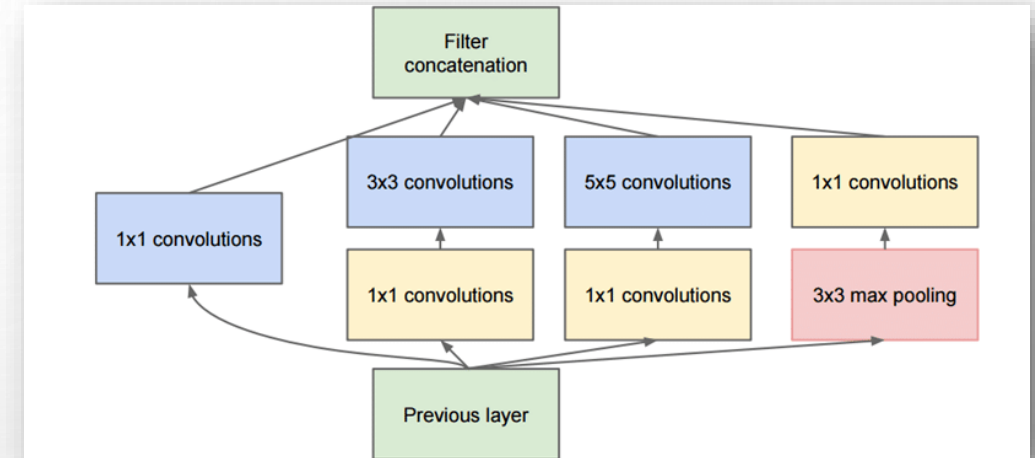


2014 – GOOGLNET / INCEPTION

Desenvolvida pela equipe do Google, a arquitetura Inception inovou ao introduzir o conceito de "blocos Inception", que aplicam convoluções de diferentes tamanhos simultaneamente.

Reduziu drasticamente o número de parâmetros, mantendo alta performance.

O GoogLeNet venceu o desafio ImageNet de 2014, com 22 camadas, mas menos parâmetros que o AlexNet.

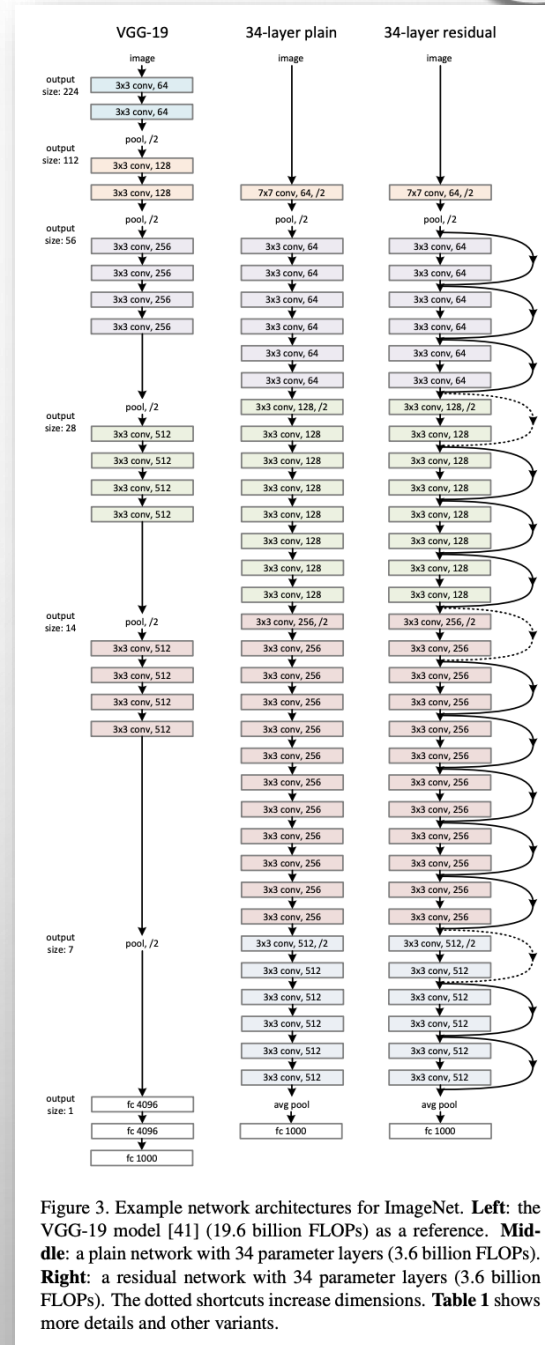


2015 - RESNET

Criada por Kaiming He e colaboradores, a ResNet introduziu a ideia de blocos residuais, que utilizam conexões de atalho (skip connections).

Esse design resolveu o problema de degradação em redes muito profundas, permitindo a criação de redes com mais de 100 camadas (como a ResNet-152).

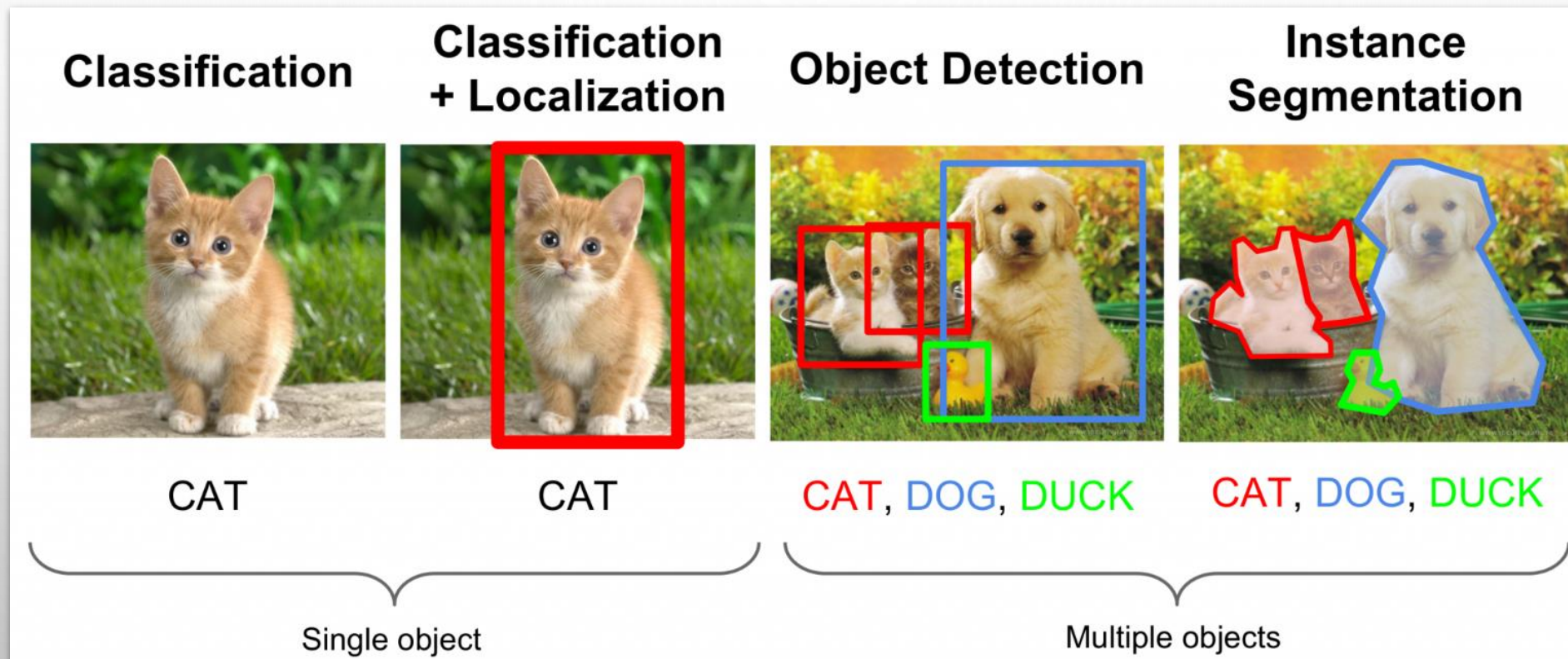
A ResNet foi outro divisor de águas ao mostrar que redes extremamente profundas podem ser treinadas com sucesso, vencendo o desafio ImageNet de 2015.



The image features a light gray background with a subtle gradient. In the top-left and bottom-right corners, there are clusters of realistic, three-dimensional water droplets of various sizes. A faint, circular, embossed-like pattern is visible in the upper center of the page, above the main text.

BUSINESS UNDERSTANDING

CLASSIFICAÇÃO DE IMAGENS

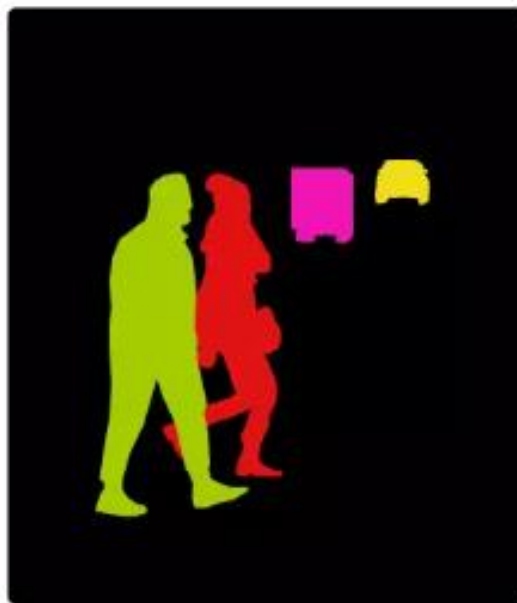


SEGMENTAÇÃO DE IMAGENS

Types of Image Segmentation



**SEMANTIC IMAGE
SEGMENTATION**

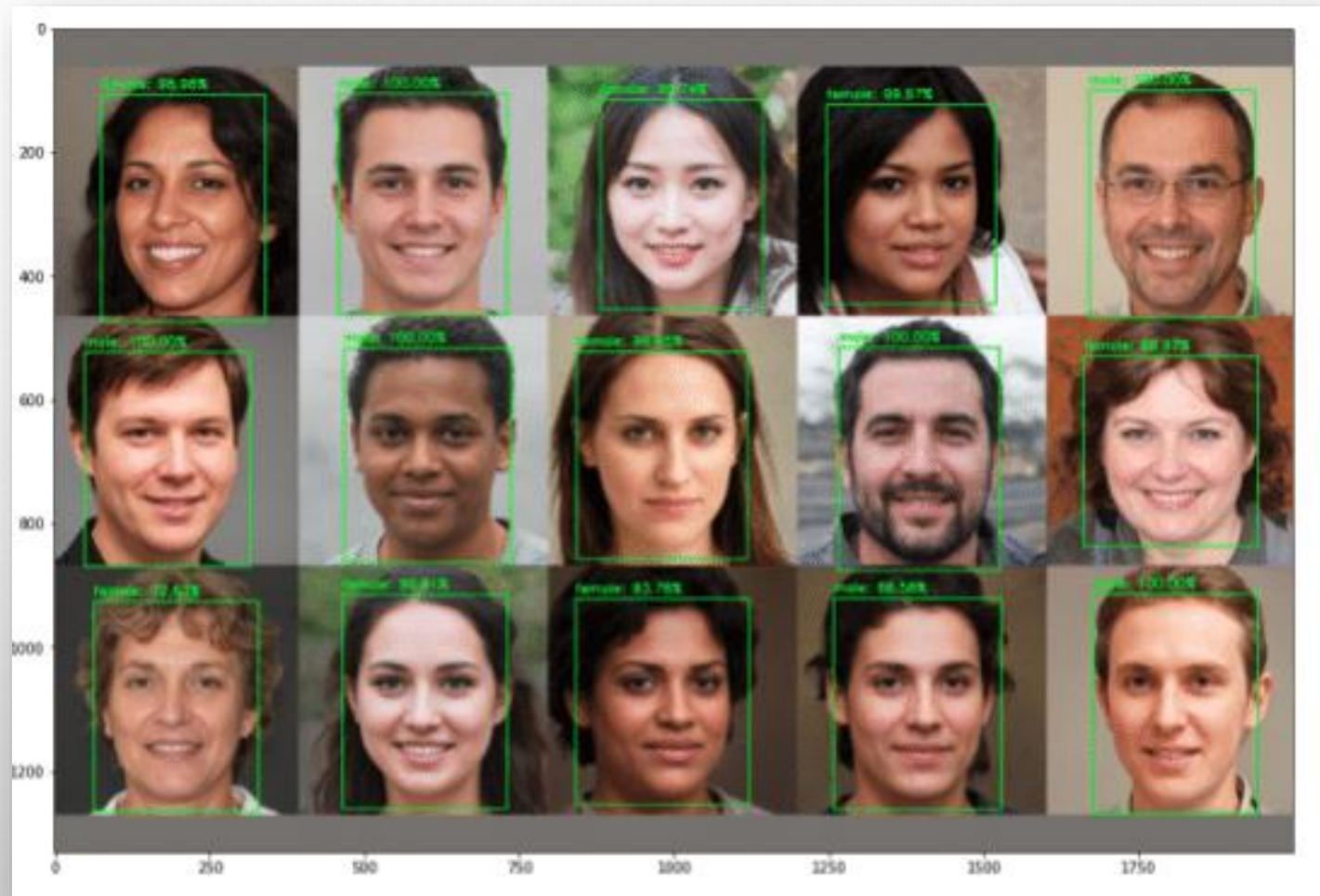


**INSTANCE
SEGMENTATION**

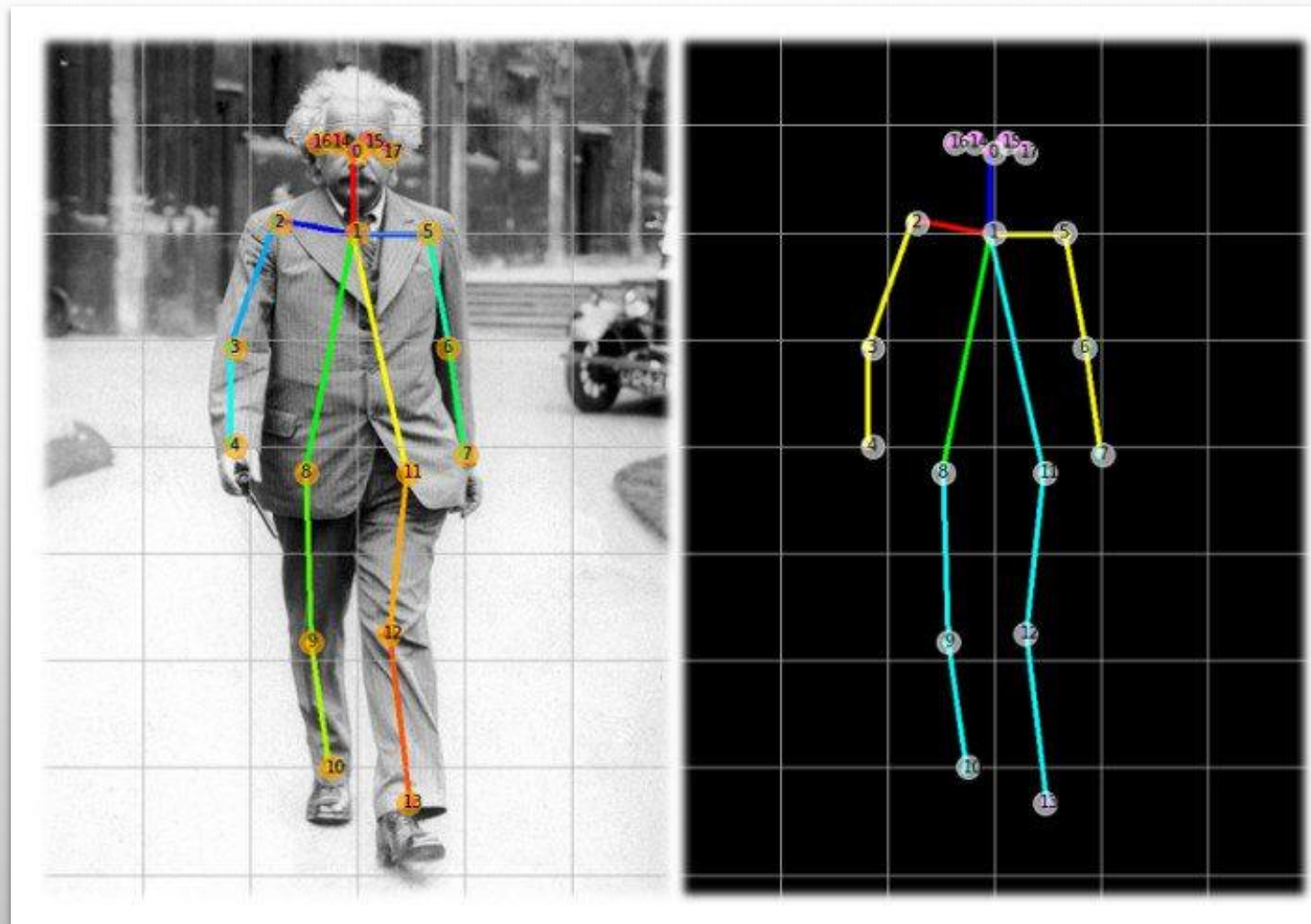


**PANOPTIC
SEGMENTATION**

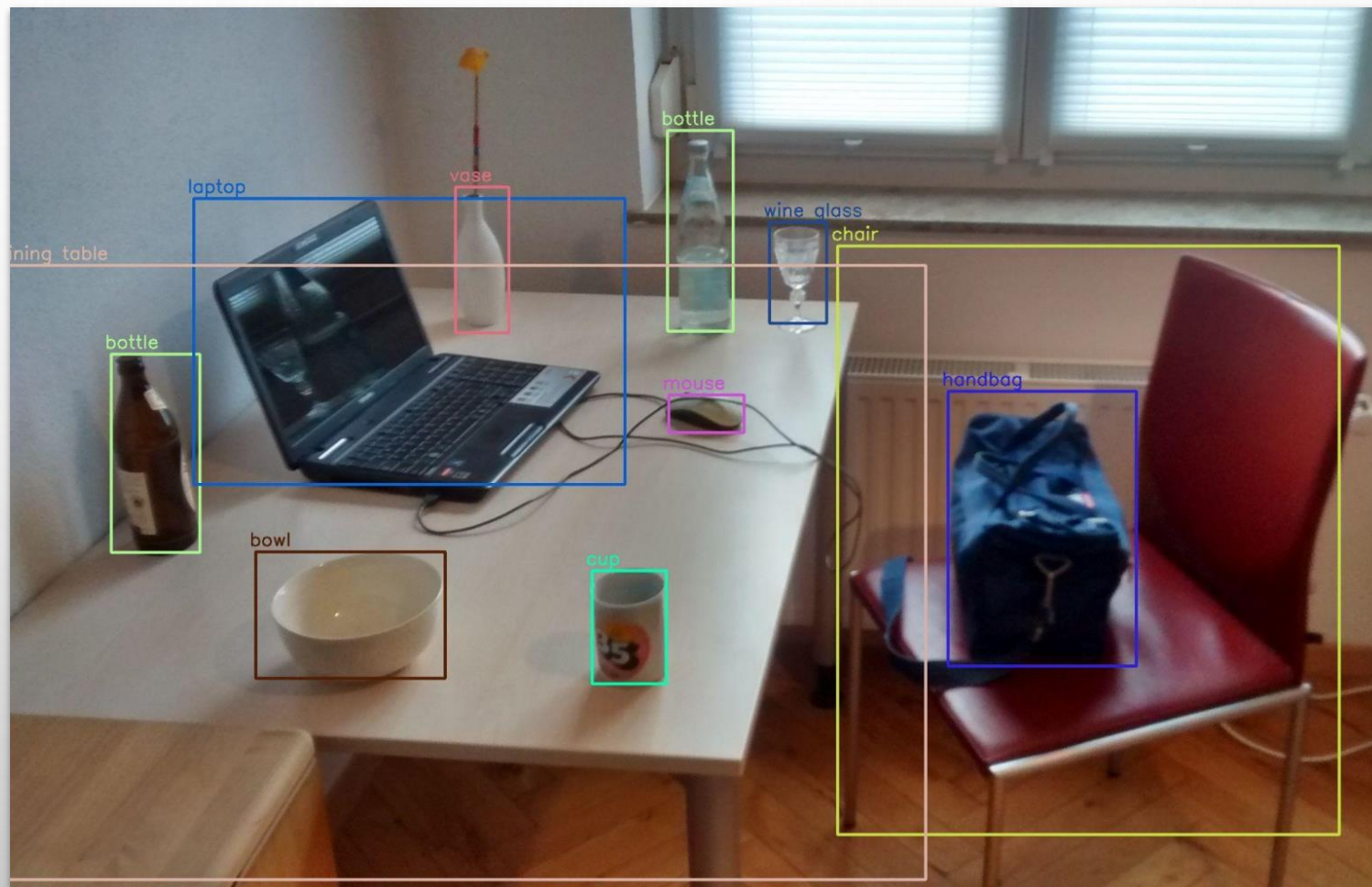
IDENTIFICAÇÃO



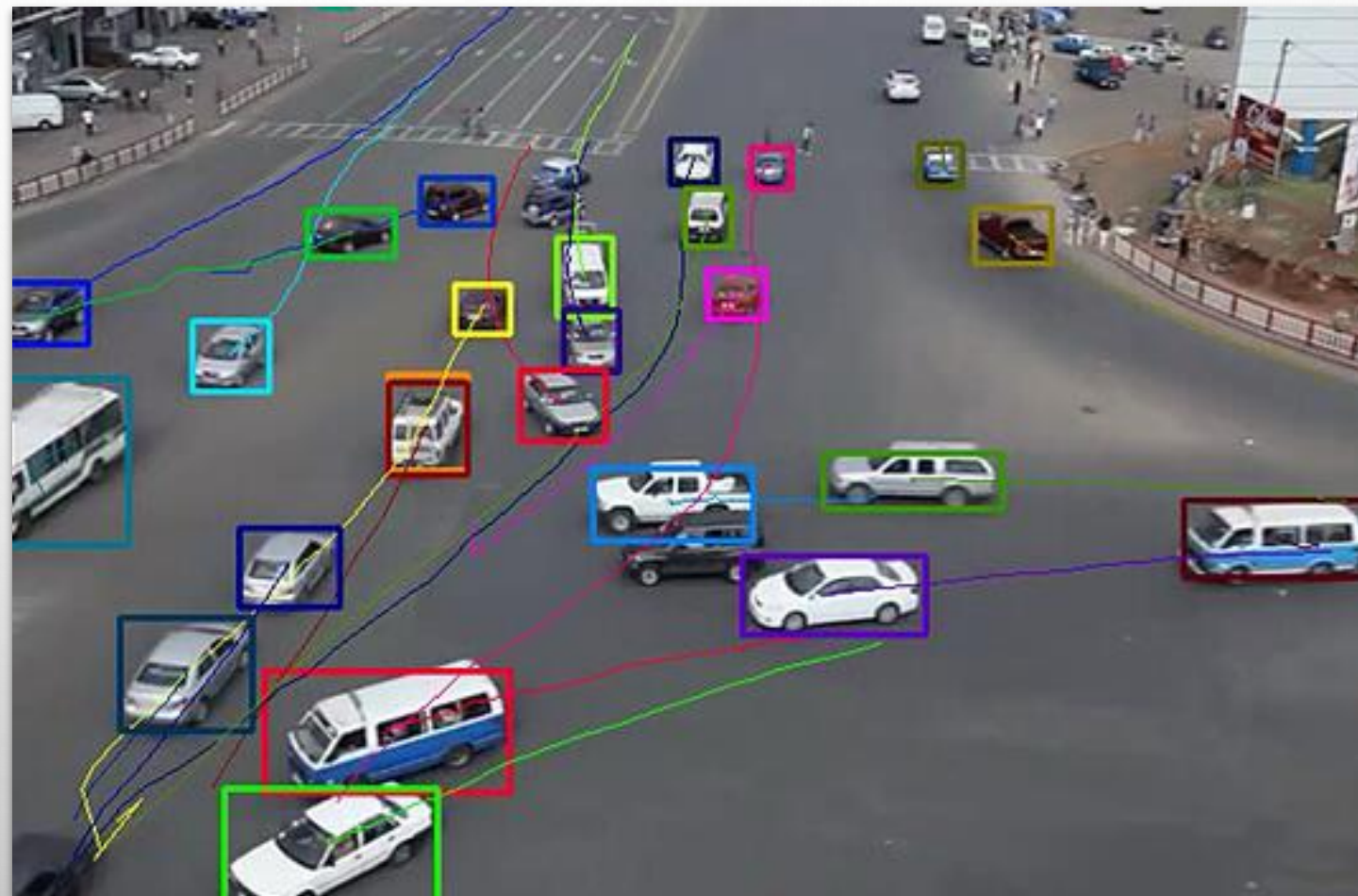
DETECÇÃO DE POSE



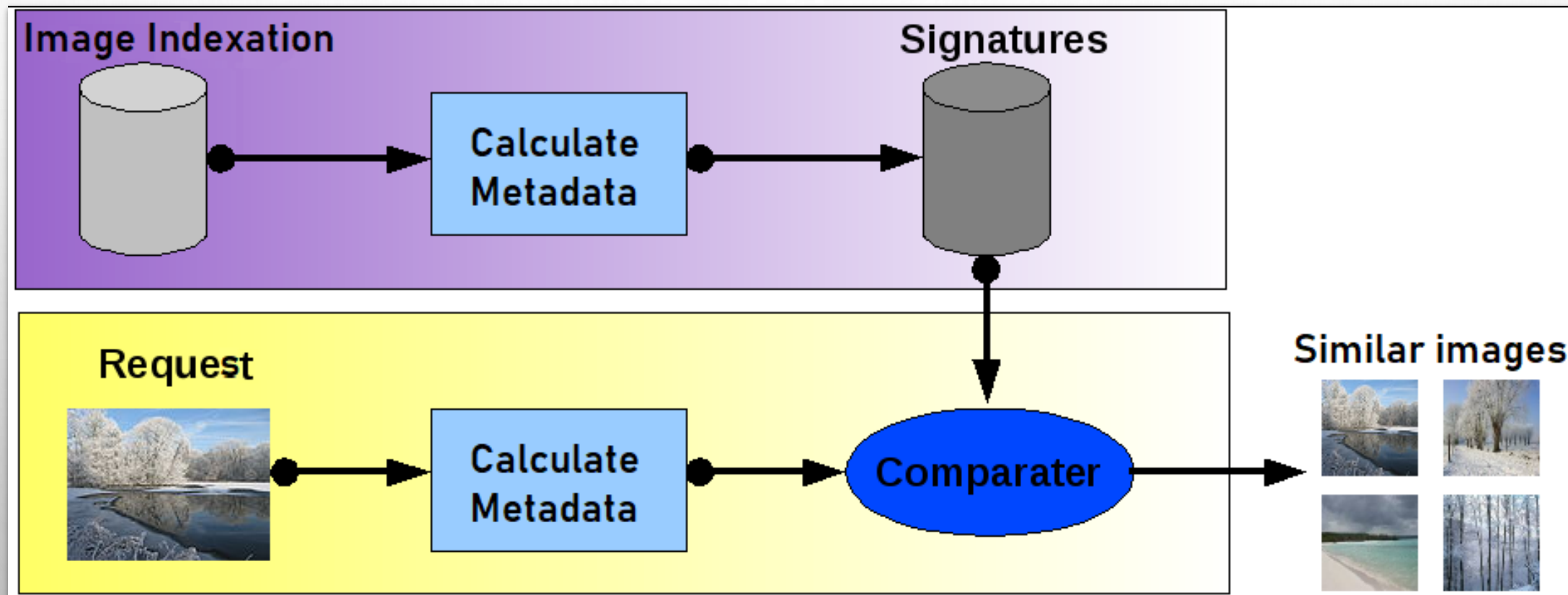
DETECÇÃO DE OBJETOS



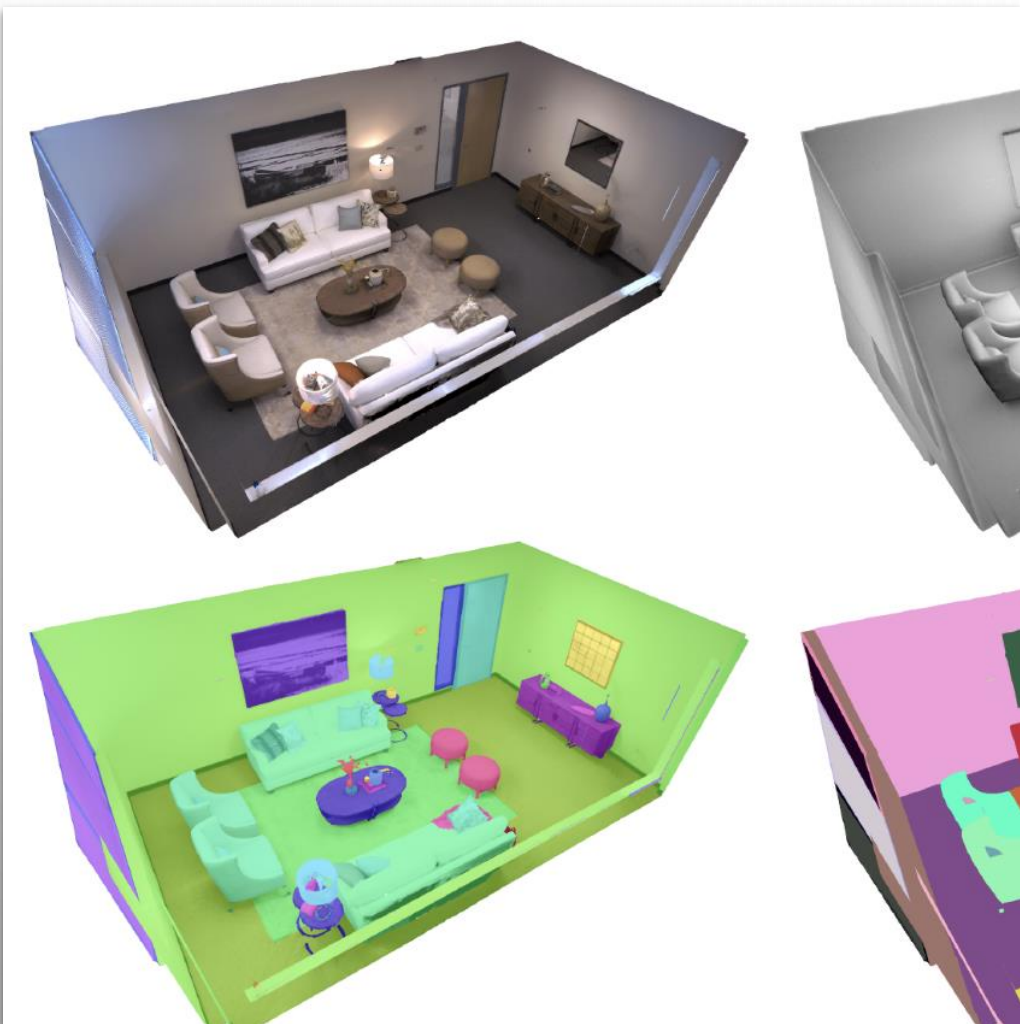
TRACKING DE OBJETOS



BUSCA INDEXADA POR IMAGEM

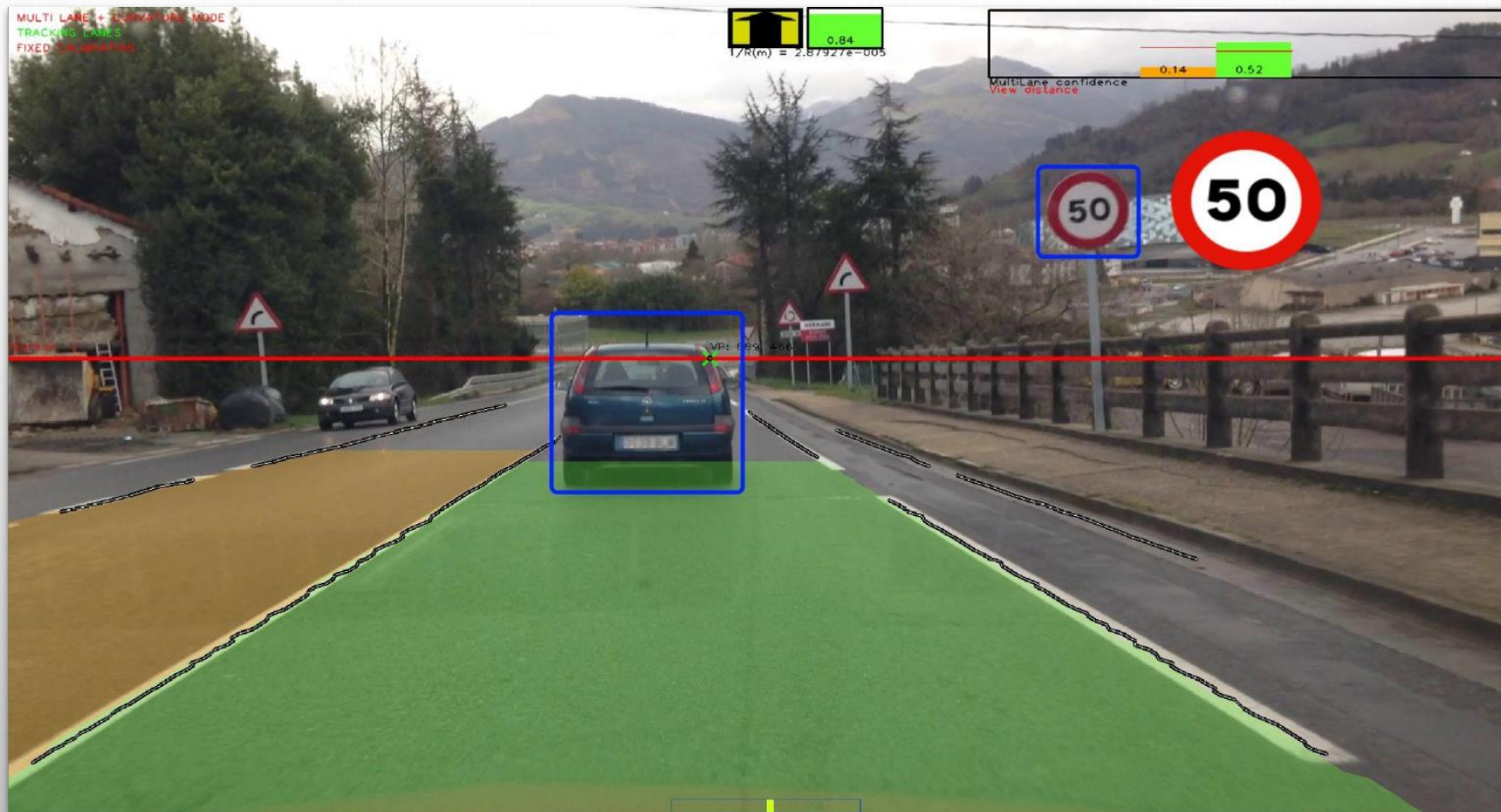


RECONSTRUÇÃO DE CENA



INSPEÇÃO AUTOMATIZADA

NAVEGAÇÃO



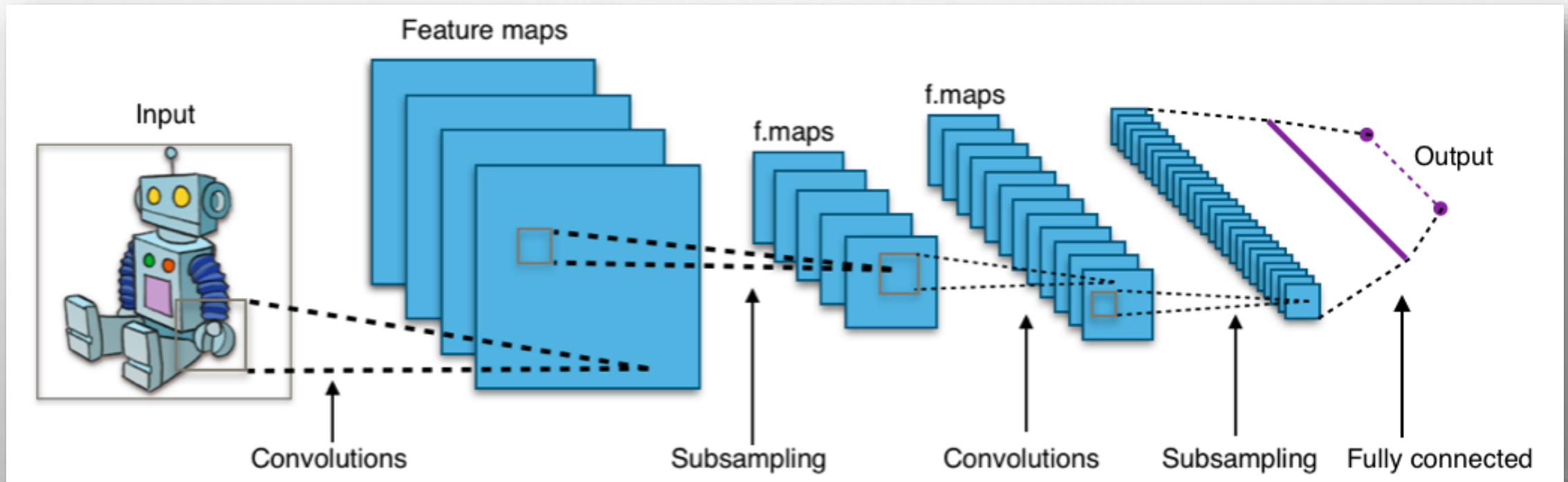
REMOÇÃO DE RUÍDO



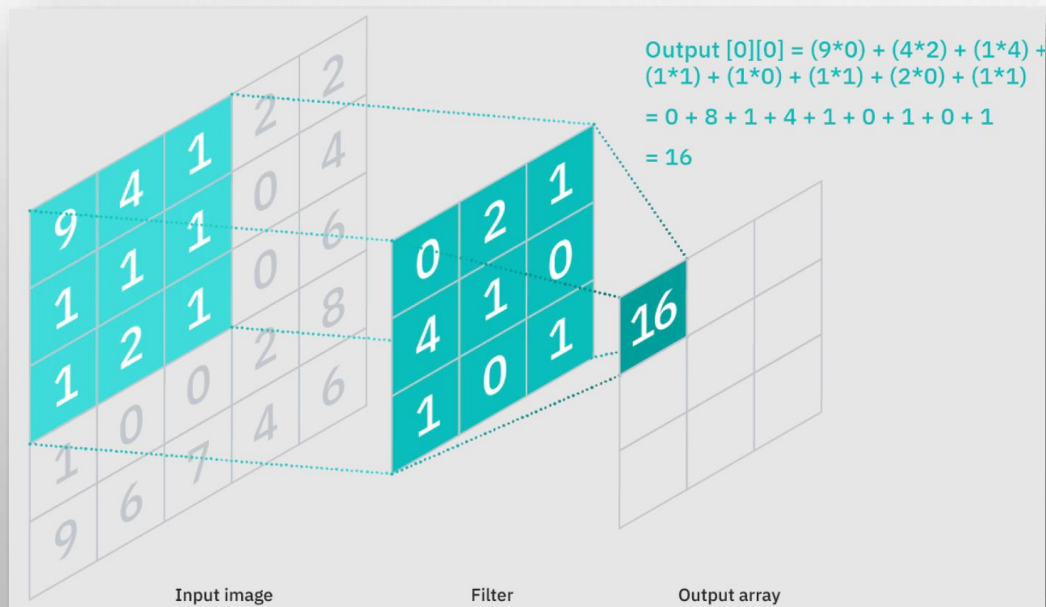
MODELOS GENERATIVOS

MODELING

REDE NEURAL CONVOLUTIVA



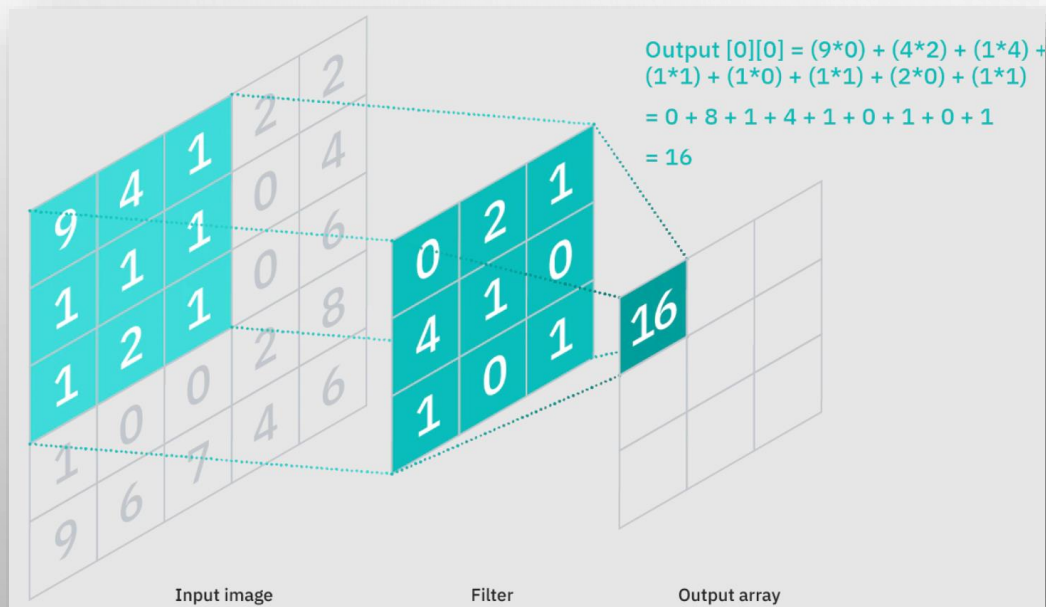
REDE NEURAL CONVOLUTIVA



Convolução

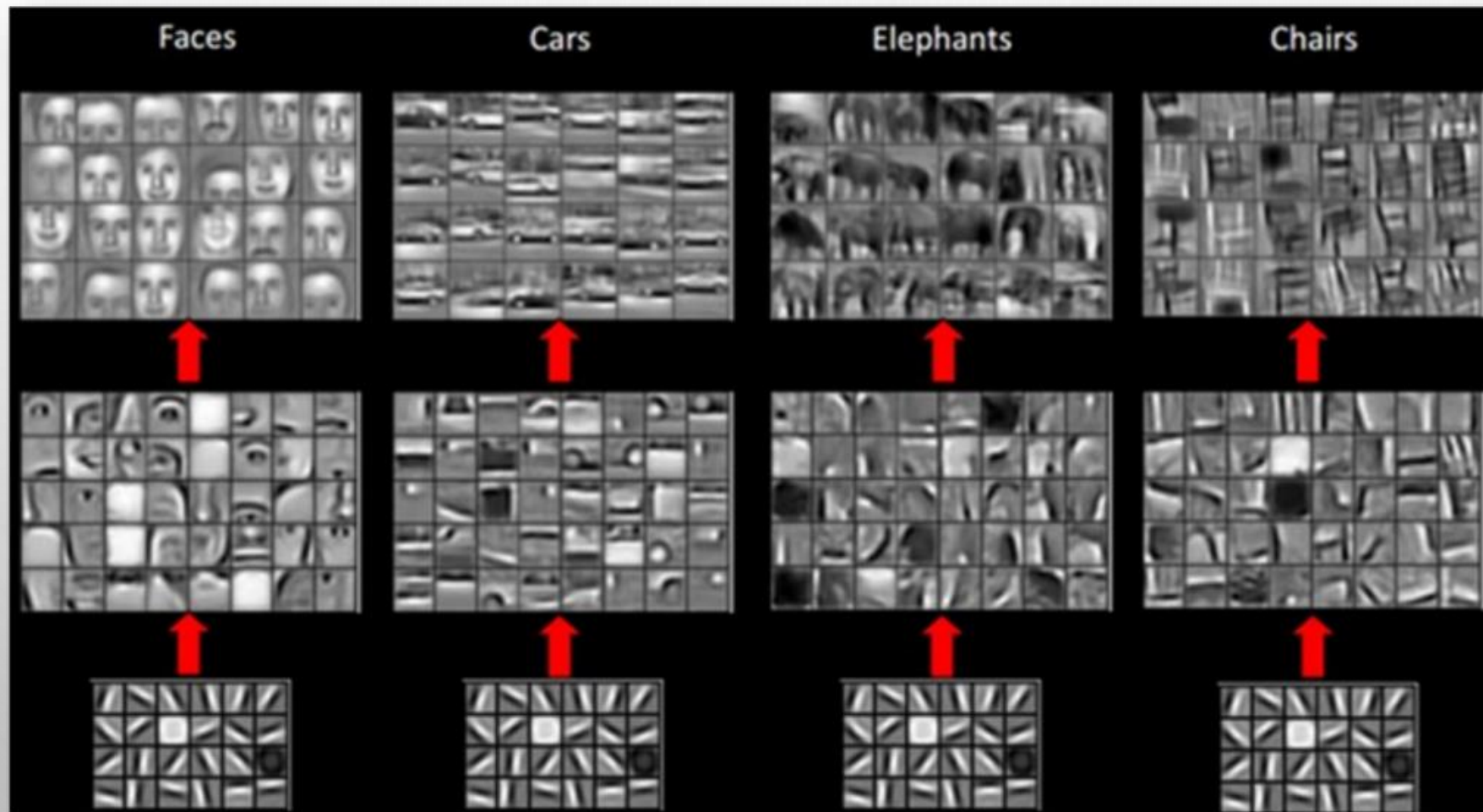
- A camada convolutiva é um dos principais blocos de construção das redes neurais convolutivas (CNNs), usada amplamente em visão computacional.
- O objetivo dessa camada é detectar padrões locais, como bordas, texturas ou formas, a partir das entradas (imagens).
- Funciona aplicando um **filtro (kernel)** sobre a imagem de entrada, que se desloca ao longo dela, realizando operações de convolução. Cada filtro aprende a detectar características específicas.

REDE NEURAL CONVOLUTIVA

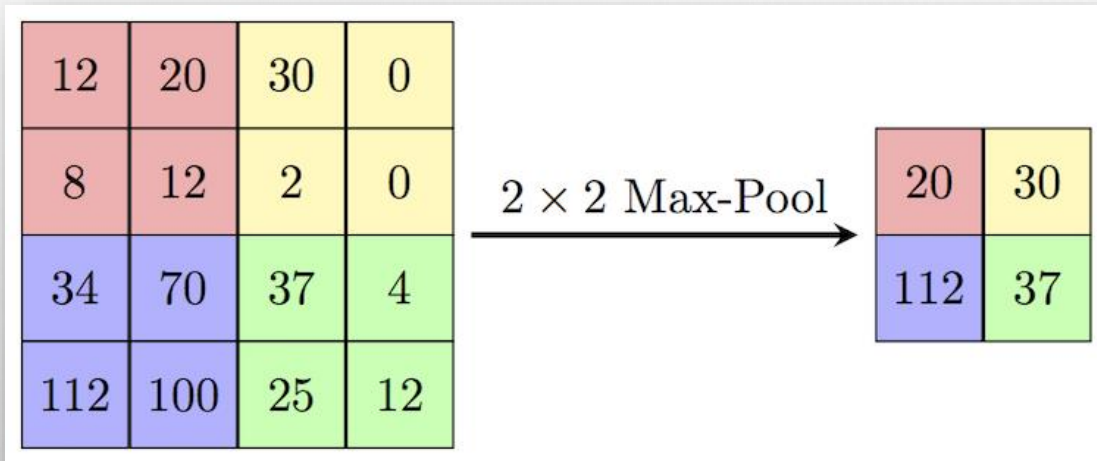


Convolução

- A operação resulta em um **mapa de características** (feature map), que destaca as características detectadas pelo filtro em diferentes regiões da imagem.
- Uma das vantagens é a **compartilhamento de pesos**, onde o mesmo filtro é usado em toda a imagem, reduzindo o número de parâmetros e tornando o modelo mais eficiente.
- Com a aplicação de múltiplos filtros, diferentes características podem ser extraídas simultaneamente, ajudando a rede a entender a imagem de forma mais completa.
- Além disso, as camadas convolutivas preservam a relação espacial dos pixels, o que é essencial para o reconhecimento de padrões complexos.



MAX POOLING



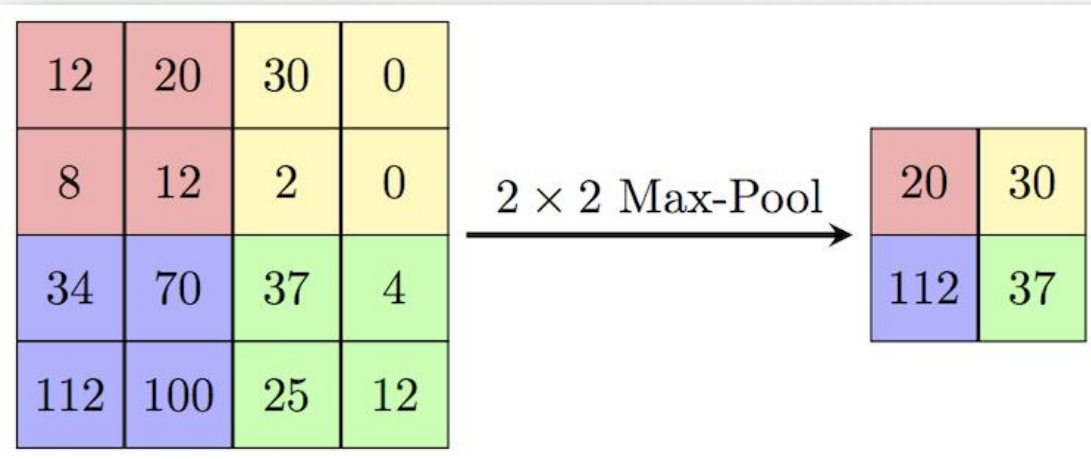
Max Pooling é uma técnica de redução de dimensionalidade usada em redes neurais convolutivas (CNNs).

Sua principal função é reduzir as dimensões espaciais (altura e largura) do mapa de características, mantendo as características mais importantes.

O pooling é aplicado em regiões da imagem e seleciona o valor máximo dentro de cada região.

Isso ajuda a tornar a rede mais eficiente ao reduzir a quantidade de dados, diminuindo a complexidade computacional e o risco de overfitting.

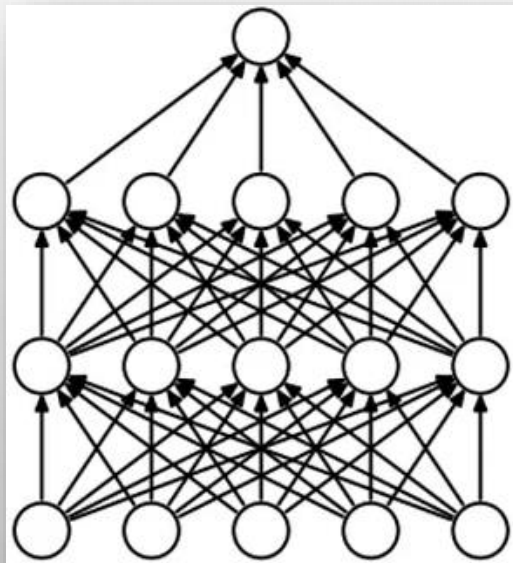
MAX POOLING



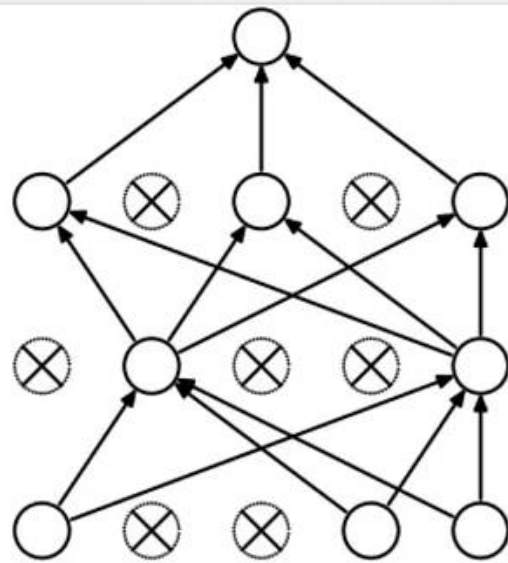
Benefícios

- **Redução de dimensionalidade:** Menos parâmetros e menos computação.
- **Invariância a pequenas translações:** Pequenas mudanças na posição do objeto na imagem não afetam a detecção.
- **Controle de overfitting:** Ao reduzir o número de neurônios, diminui-se o risco de overfitting.

REDE NEURAL CONVOLUTIVA

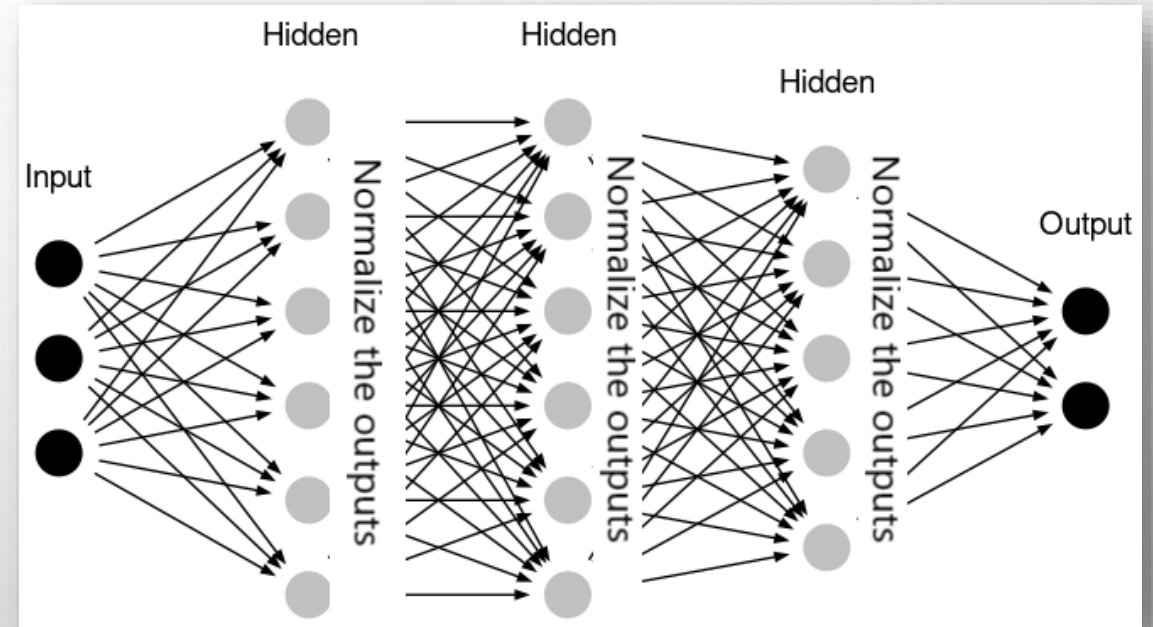


(a) Standard Neural Net



(b) After applying dropout.

Dropout



Batch Normalization

The background is a light gray gradient. In the top-left and bottom-right corners, there are several realistic water droplets of various sizes, some overlapping. A faint, circular watermark is visible in the upper center of the page.

PARTE 2 : PRÁTICA

AMBIENTE PYTHON



6. Machine Learning

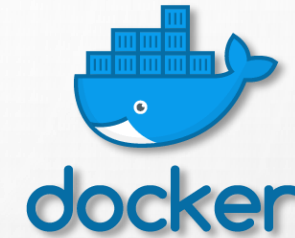


6. Deployment



4. Variáveis Aleatórias

5. Visualização



1. Editor de Código



2. Gestor de Ambiente



3. Ambiente Python do Projeto



3. Notebook Dinâmico

WORKSHOP

- ~~• QUAL A TOPOLOGIA DE DEEP LEARNING ADEQUADA PARA O MEU TRABALHO?~~
- ~~• QUAL CAPÍTULO DO LIVRO MELHOR SE ENQUADRA NO MEU TRABALHO?~~
- **AULA 3: NOVO CICLO DE BUSINESS UNDERSTANDING / GRUPO + MODELO BASELINE TREINADO**
- AULA 5 OU 7: MODELO PROFUNDO TREINADO
- AULA 7: DEPLOYMENT DO MODELO*
- AULA 3-7 > APRESENTAÇÃO TEÓRICA DA(S) TOPOLOGIA(S) + LEITURA DE ARTIGO + ACOMPANHAMENTO DOS TRABALHOS + DEEP DIVE NO CÓDIGO (POR GRUPO)
- APRESENTAÇÃO FINAL DOS TRABALHOS

**Passo Opcional*